ITU COMPUTER ENGINEERING DEPARTMENT

BLG 233E DATA STRUCTURES

HOMEWORK -1

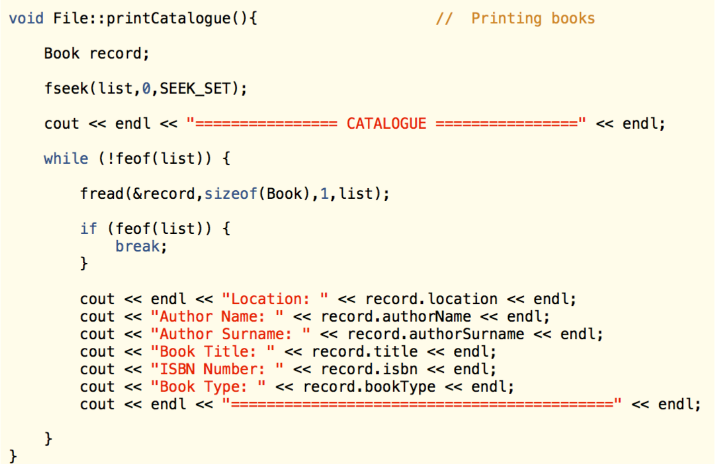
Name, Surname: Can Yılmaz Altıniğne

Student Number: 150130132

REPORT

I used XCode 7 IDE and GCC 4.2.1 Compiler to write and compile the codes. I created 1 Header File to define structs and 2 C++ Files one of them for functions in struct and the other one for the main program. As data structures, I used file operations and struct for homework. Using file operations prevented the record loss so I kept records in a text file and it provides me that when I open the program again I can find old records.

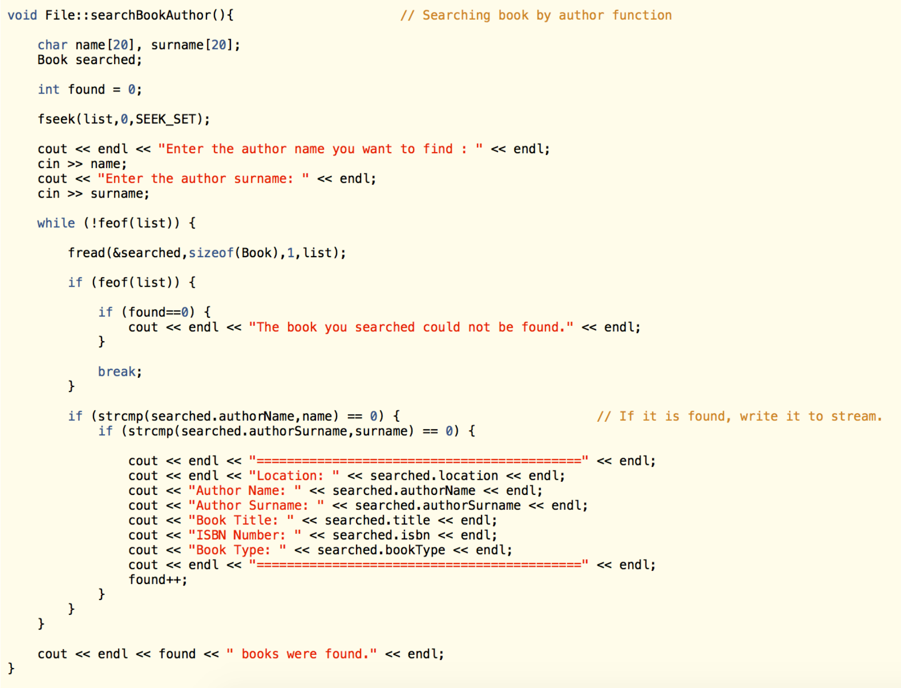
Printing (P)



```
void File::printCatalogue(){                    //  Printing books

    Book record;

    fseek(list,0,SEEK_SET);

    cout << endl << "================ CATALOGUE ================" << endl;

    while (!feof(list)) {

        fread(&record,sizeof(Book),1,list);

        if (feof(list)) {
            break;
        }

        cout << endl << "Location: " << record.location << endl;
        cout << "Author Name: " << record.authorName << endl;
        cout << "Author Surname: " << record.authorSurname << endl;
        cout << "Book Title: " << record.title << endl;
        cout << "ISBN Number: " << record.isbn << endl;
        cout << "Book Type: " << record.bookType << endl;
        cout << endl << "============================================" << endl;

    }
}
```

The records are read from list to the end of file and if there are records, they are written on console.

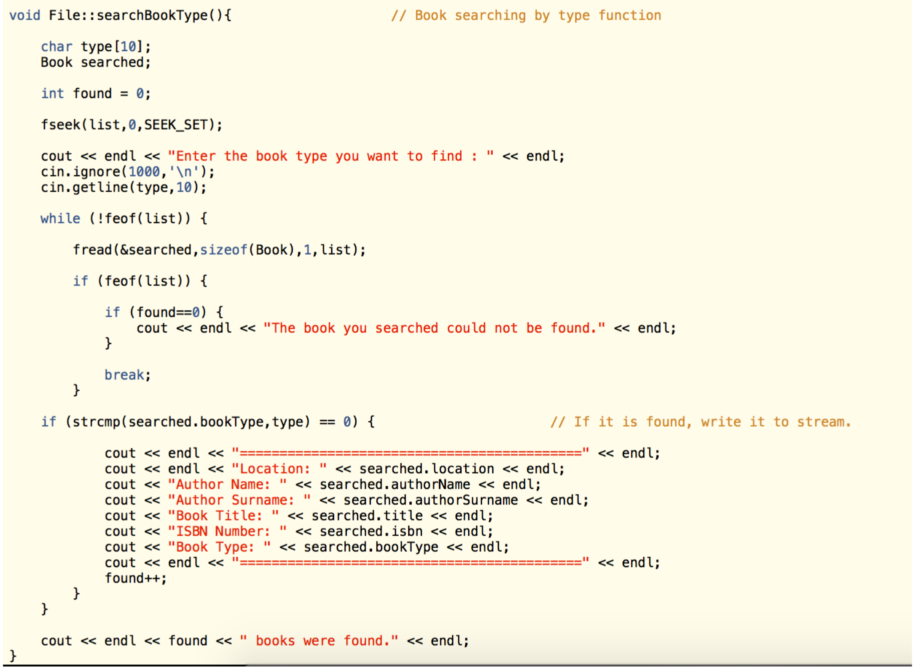
Searching by author (A)



```
void File::searchBookAuthor(){                          // Searching book by author function

    char name[20], surname[20];
    Book searched;

    int found = 0;

    fseek(list,0,SEEK_SET);

    cout << endl << "Enter the author name you want to find : " << endl;
    cin >> name;
    cout << "Enter the author surname: " << endl;
    cin >> surname;

    while (!feof(list)) {

        fread(&searched,sizeof(Book),1,list);

        if (feof(list)) {

            if (found==0) {
                cout << endl << "The book you searched could not be found." << endl;
            }

            break;
        }

        if (strcmp(searched.authorName,name) == 0) {                          // If it is found, write it to stream.
            if (strcmp(searched.authorSurname,surname) == 0) {

                cout << endl << "============================================" << endl;
                cout << endl << "Location: " << searched.location << endl;
                cout << "Author Name: " << searched.authorName << endl;
                cout << "Author Surname: " << searched.authorSurname << endl;
                cout << "Book Title: " << searched.title << endl;
                cout << "ISBN Number: " << searched.isbn << endl;
                cout << "Book Type: " << searched.bookType << endl;
                cout << endl << "============================================" << endl;
                found++;
            }
        }
    }

    cout << endl << found << " books were found." << endl;
}
```

Name and surname are entered by user. If there are records, they are assigned to “searched” variable respectively and if the entered name and surname equal to found name and surname on the list, the records is written on console.

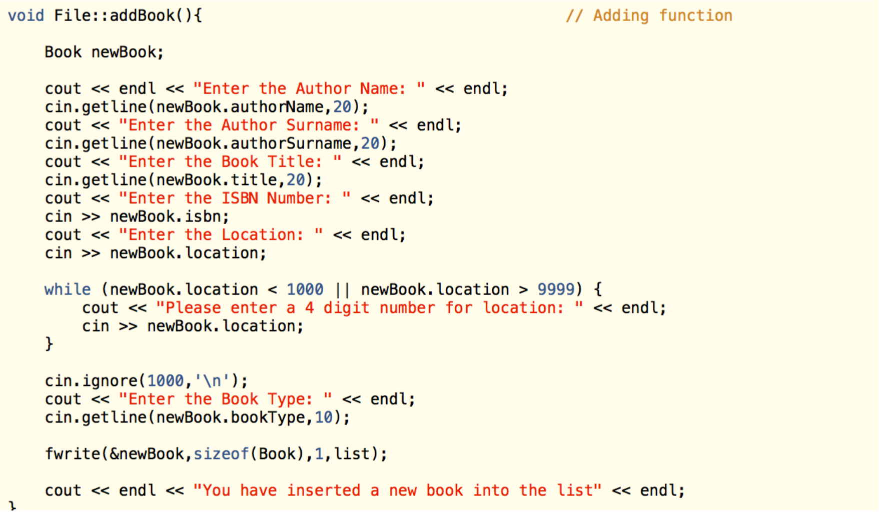
Searching by type (T)



```
void File::searchBookType(){                          // Book searching by type function

    char type[10];
    Book searched;

    int found = 0;

    fseek(list,0,SEEK_SET);

    cout << endl << "Enter the book type you want to find : " << endl;
    cin.ignore(1000,'\n');
    cin.getline(type,10);

    while (!feof(list)) {

        fread(&searched,sizeof(Book),1,list);

        if (feof(list)) {

            if (found==0) {
                cout << endl << "The book you searched could not be found." << endl;
            }

            break;
        }

    if (strcmp(searched.bookType,type) == 0) {                          // If it is found, write it to stream.

            cout << endl << "============================================" << endl;
            cout << endl << "Location: " << searched.location << endl;
            cout << "Author Name: " << searched.authorName << endl;
            cout << "Author Surname: " << searched.authorSurname << endl;
            cout << "Book Title: " << searched.title << endl;
            cout << "ISBN Number: " << searched.isbn << endl;
            cout << "Book Type: " << searched.bookType << endl;
            cout << endl << "============================================" << endl;
            found++;
        }
    }

    cout << endl << found << " books were found." << endl;
}
```

It is very similar to searching by author function but this time the type of book is entered.

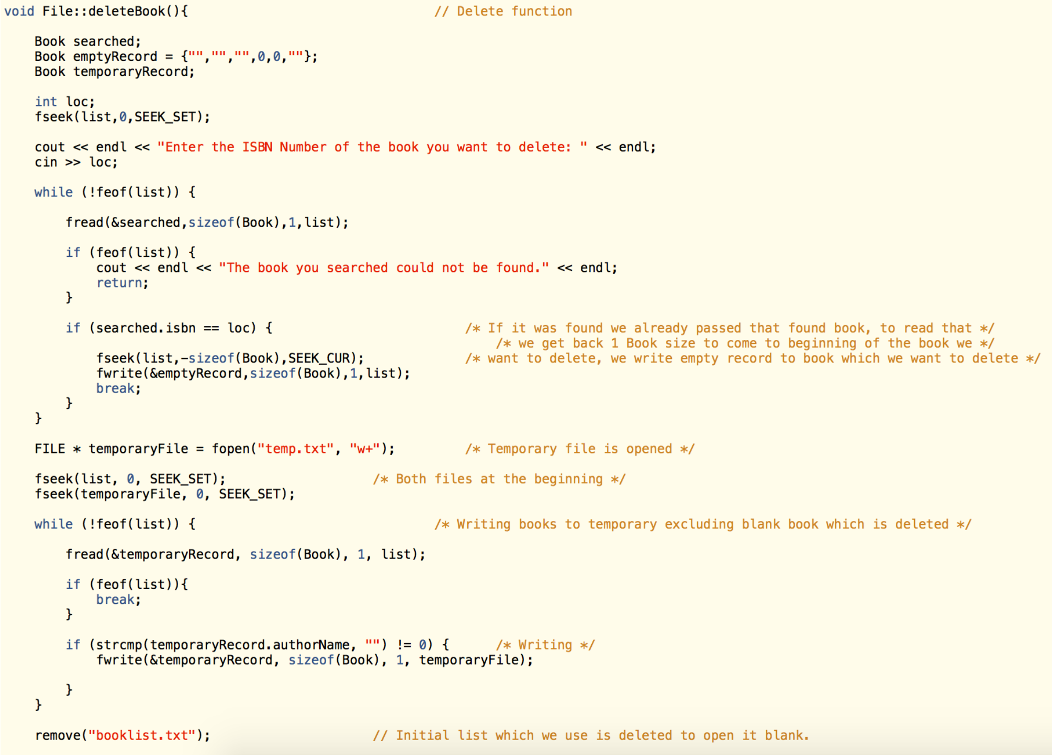
Insert (I)



```
void File::addBook(){                                           // Adding function

    Book newBook;

    cout << endl << "Enter the Author Name: " << endl;
    cin.getline(newBook.authorName,20);
    cout << "Enter the Author Surname: " << endl;
    cin.getline(newBook.authorSurname,20);
    cout << "Enter the Book Title: " << endl;
    cin.getline(newBook.title,20);
    cout << "Enter the ISBN Number: " << endl;
    cin >> newBook.isbn;
    cout << "Enter the Location: " << endl;
    cin >> newBook.location;

    while (newBook.location < 1000 || newBook.location > 9999) {
        cout << "Please enter a 4 digit number for location: " << endl;
        cin >> newBook.location;
    }

    cin.ignore(1000,'\n');
    cout << "Enter the Book Type: " << endl;
    cin.getline(newBook.bookType,10);

    fwrite(&newBook,sizeof(Book),1,list);

    cout << endl << "You have inserted a new book into the list" << endl;
}
```

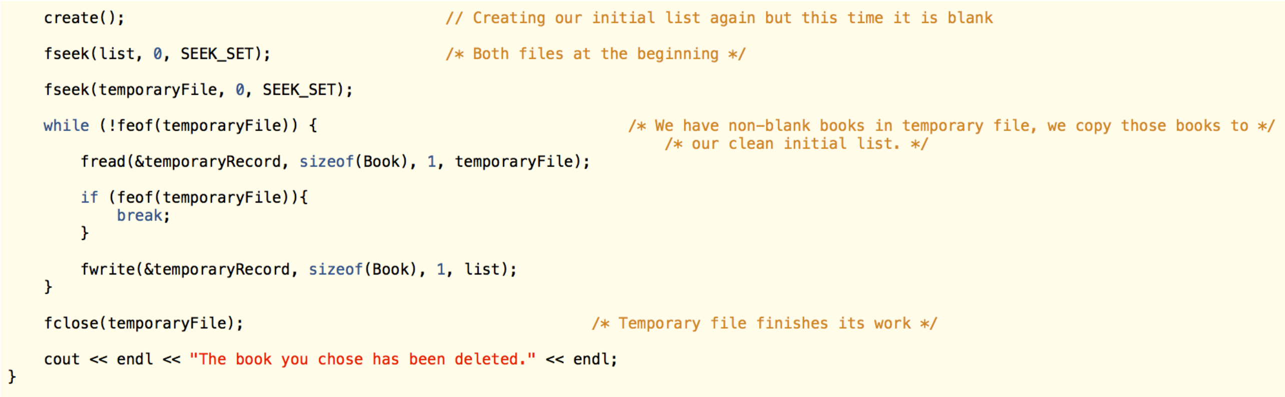
Variables are entered by user. The entries for location are checked if they are 4-Digit number or not. Finally new entry is added to list which is text file opened at the beginning of the program.

Remove (R)



```
void File::deleteBook(){                                    // Delete function

    Book searched;
    Book emptyRecord = {"","","",0,0,""};
    Book temporaryRecord;

    int loc;
    fseek(list,0,SEEK_SET);

    cout << endl << "Enter the ISBN Number of the book you want to delete: " << endl;
    cin >> loc;

    while (!feof(list)) {

        fread(&searched,sizeof(Book),1,list);

        if (feof(list)) {
            cout << endl << "The book you searched could not be found." << endl;
            return;
        }

        if (searched.isbn == loc) {                         /* If it was found we already passed that found book, to read that */
                                                                /* we get back 1 Book size to come to beginning of the book we */
            fseek(list,-sizeof(Book),SEEK_CUR);             /* want to delete, we write empty record to book which we want to delete */
            fwrite(&emptyRecord,sizeof(Book),1,list);
            break;
        }
    }

    FILE * temporaryFile = fopen("temp.txt", "w+");         /* Temporary file is opened */

    fseek(list, 0, SEEK_SET);                          /* Both files at the beginning */
    fseek(temporaryFile, 0, SEEK_SET);

    while (!feof(list)) {                                   /* Writing books to temporary excluding blank book which is deleted */

        fread(&temporaryRecord, sizeof(Book), 1, list);

        if (feof(list)){
            break;
        }

        if (strcmp(temporaryRecord.authorName, "") != 0) {      /* Writing */
            fwrite(&temporaryRecord, sizeof(Book), 1, temporaryFile);

        }
    }

    remove("booklist.txt");                            // Initial list which we use is deleted to open it blank.
```

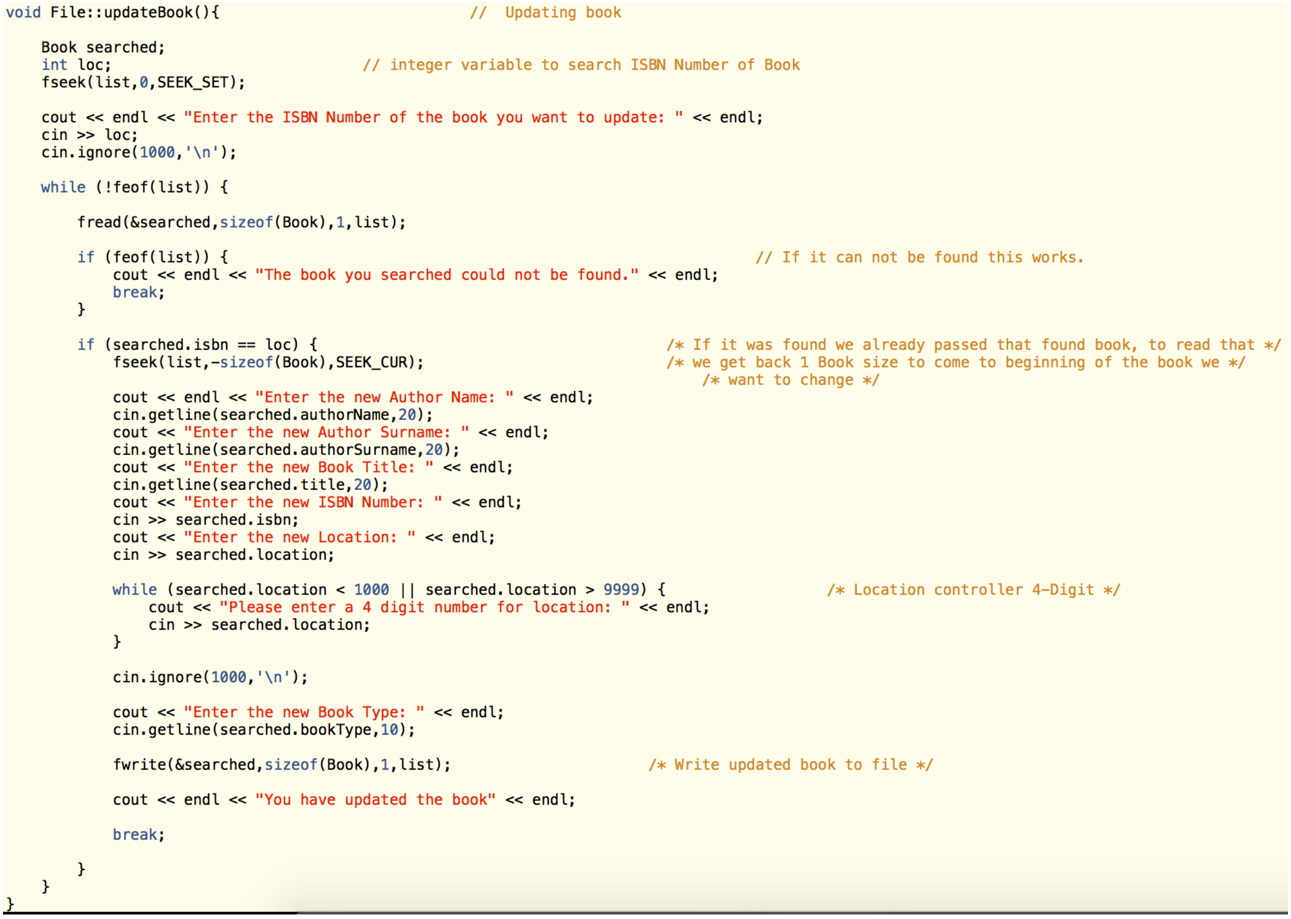
ISBN number of the book is entered. If it can not be found function ends with return command. When it is found a new temporary text file is created. In the original text file, the cursor is brought to beginning of the record which is going to be deleted and empty record is written in place of this record. Contents of the original text file is written to temporary file (Empty records are not written). Then our original text “booklist.txt” is removed.



```
    create();                                   // Creating our initial list again but this time it is blank

    fseek(list, 0, SEEK_SET);                   /* Both files at the beginning */

    fseek(temporaryFile, 0, SEEK_SET);

    while (!feof(temporaryFile)) {                                  /* We have non-blank books in temporary file, we copy those books to */
                                                                        /* our clean initial list. */
        fread(&temporaryRecord, sizeof(Book), 1, temporaryFile);

        if (feof(temporaryFile)){
            break;
        }

        fwrite(&temporaryRecord, sizeof(Book), 1, list);
    }

    fclose(temporaryFile);                                      /* Temporary file finishes its work */

    cout << endl << "The book you chose has been deleted." << endl;
}
```

After deleting original text file, it is created again as a empty text file with create( ) function. Contents of the temporary file is written to original file and then temporary file is closed. Now, the record which was wanted to delete is not on the list anymore.

Update (U)

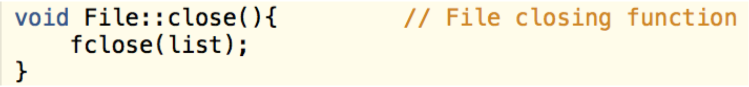


```
void File::updateBook(){                              //  Updating book

    Book searched;
    int loc;                                // integer variable to search ISBN Number of Book
    fseek(list,0,SEEK_SET);

    cout << endl << "Enter the ISBN Number of the book you want to update: " << endl;
    cin >> loc;
    cin.ignore(1000,'\n');

    while (!feof(list)) {

        fread(&searched,sizeof(Book),1,list);

        if (feof(list)) {                                               // If it can not be found this works.
            cout << endl << "The book you searched could not be found." << endl;
            break;
        }

        if (searched.isbn == loc) {                             /* If it was found we already passed that found book, to read that */
            fseek(list,-sizeof(Book),SEEK_CUR);                 /* we get back 1 Book size to come to beginning of the book we */
                                                                    /* want to change */

            cout << endl << "Enter the new Author Name: " << endl;
            cin.getline(searched.authorName,20);
            cout << "Enter the new Author Surname: " << endl;
            cin.getline(searched.authorSurname,20);
            cout << "Enter the new Book Title: " << endl;
            cin.getline(searched.title,20);
            cout << "Enter the new ISBN Number: " << endl;
            cin >> searched.isbn;
            cout << "Enter the new Location: " << endl;
            cin >> searched.location;

            while (searched.location < 1000 || searched.location > 9999) {              /* Location controller 4-Digit */
                cout << "Please enter a 4 digit number for location: " << endl;
                cin >> searched.location;
            }

            cin.ignore(1000,'\n');

            cout << "Enter the new Book Type: " << endl;
            cin.getline(searched.bookType,10);

            fwrite(&searched,sizeof(Book),1,list);                  /* Write updated book to file */

            cout << endl << "You have updated the book" << endl;

            break;

        }
    }
}
```

ISBN Number of the book is entered. The record is searched to the end of the file. If it can not be found with break command while loop is ended. If it is found, the cursor is brought to beginning of that record and it is wanted from the user to enter new values to assign to variables. Then the new values written to the original text file.

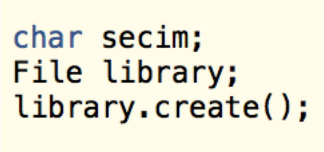
Exit (E)

This closes the original text file.

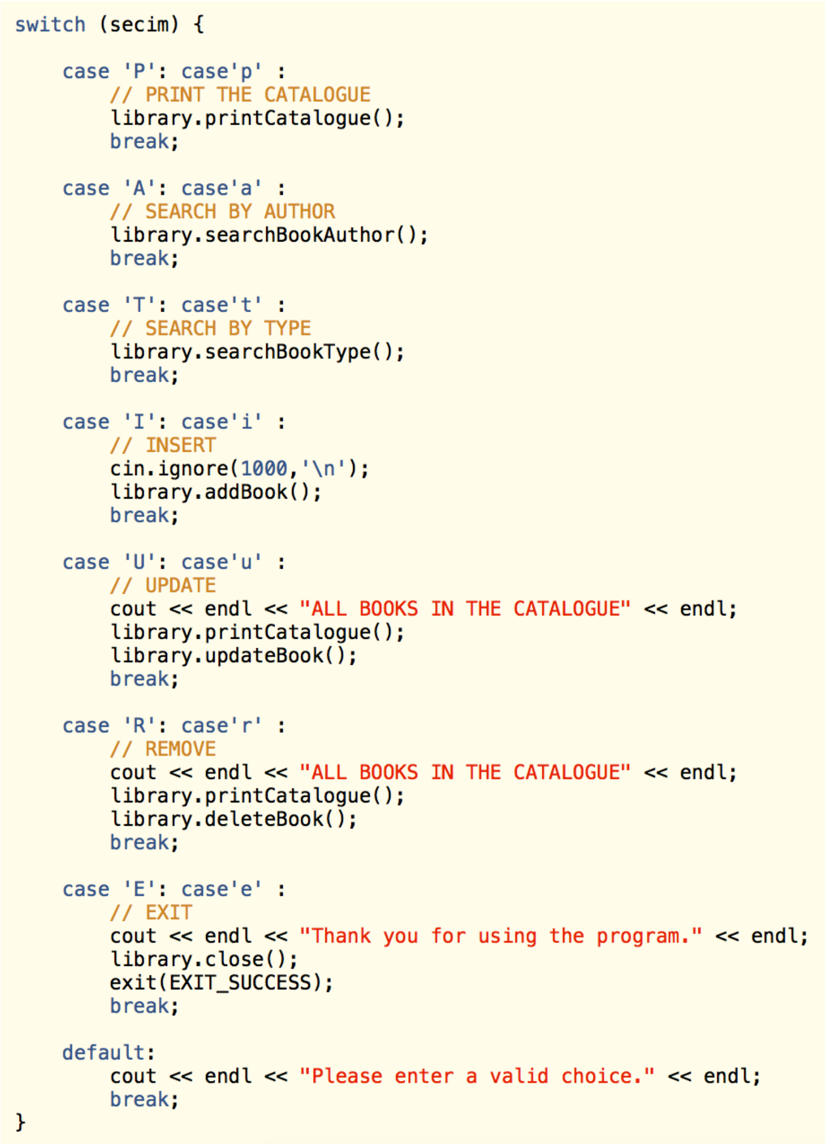


```
void File::close(){          // File closing function
    fclose(list);
}
```

Main Function



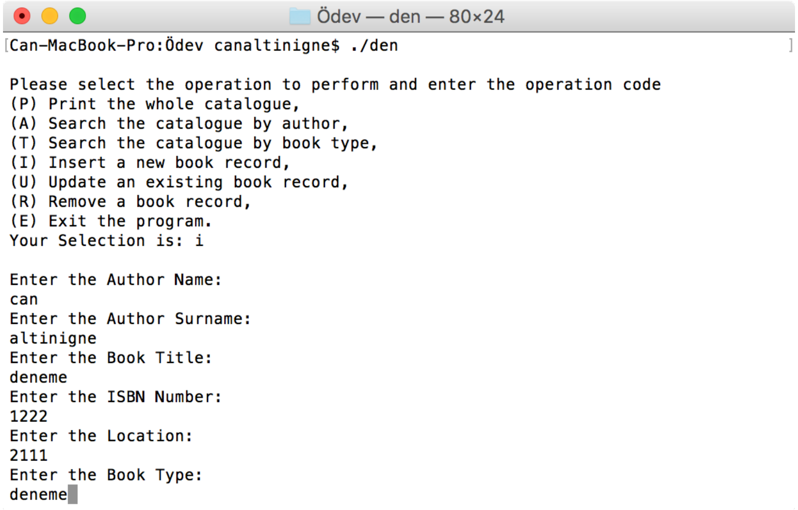
```
char secim;
File library;
library.create();
```



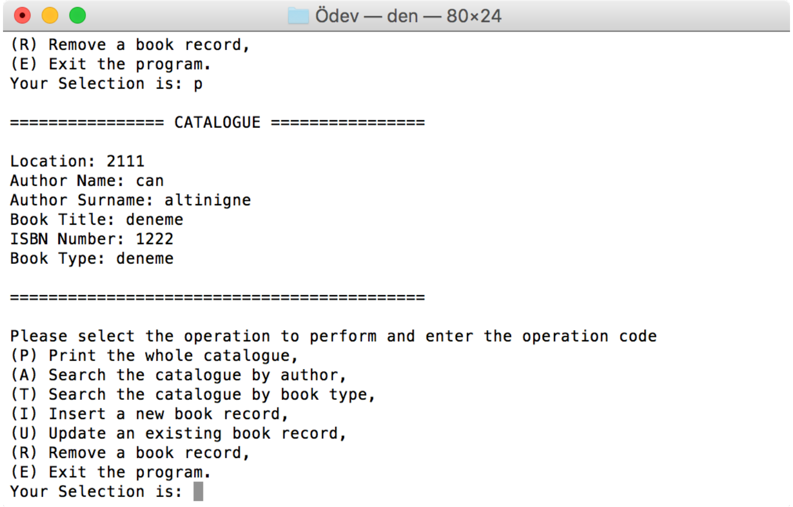
```
switch (secim) {

    case 'P': case'p' :
        // PRINT THE CATALOGUE
        library.printCatalogue();
        break;

    case 'A': case'a' :
        // SEARCH BY AUTHOR
        library.searchBookAuthor();
        break;

    case 'T': case't' :
        // SEARCH BY TYPE
        library.searchBookType();
        break;

    case 'I': case'i' :
        // INSERT
        cin.ignore(1000,'\n');
        library.addBook();
        break;

    case 'U': case'u' :
        // UPDATE
        cout << endl << "ALL BOOKS IN THE CATALOGUE" << endl;
        library.printCatalogue();
        library.updateBook();
        break;

    case 'R': case'r' :
        // REMOVE
        cout << endl << "ALL BOOKS IN THE CATALOGUE" << endl;
        library.printCatalogue();
        library.deleteBook();
        break;

    case 'E': case'e' :
        // EXIT
        cout << endl << "Thank you for using the program." << endl;
        library.close();
        exit(EXIT_SUCCESS);
        break;

    default:
        cout << endl << "Please enter a valid choice." << endl;
        break;
}
```

Before main( ) function a char variable is created to find choice of user and a File variable for records. With create function new text file is opened to save records. In a infinite while loop (while(1)) the main menu is showed and it is asked to enter a character. When the user enters a char value the switch is executed. In update and delete functions, the catalogue is written on console before the user enters the ISBN value of the book. In exit function, the original text file is closed with close( ) and exit function terminates the program.

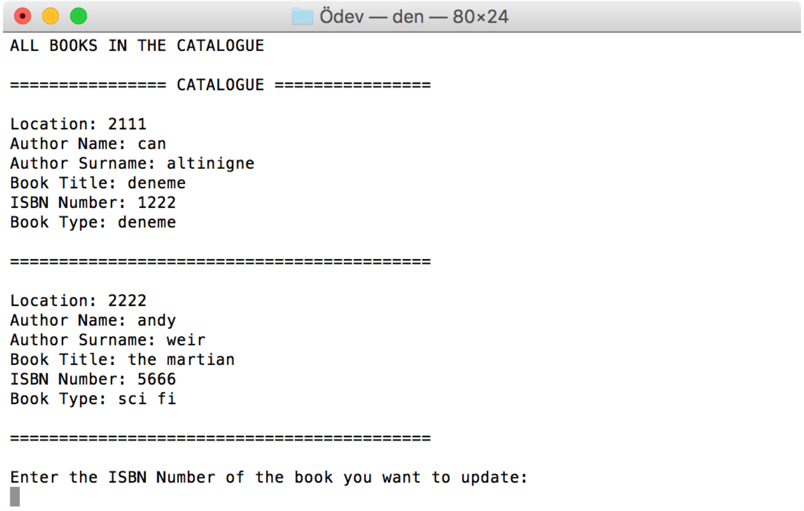
Screenshot of the console



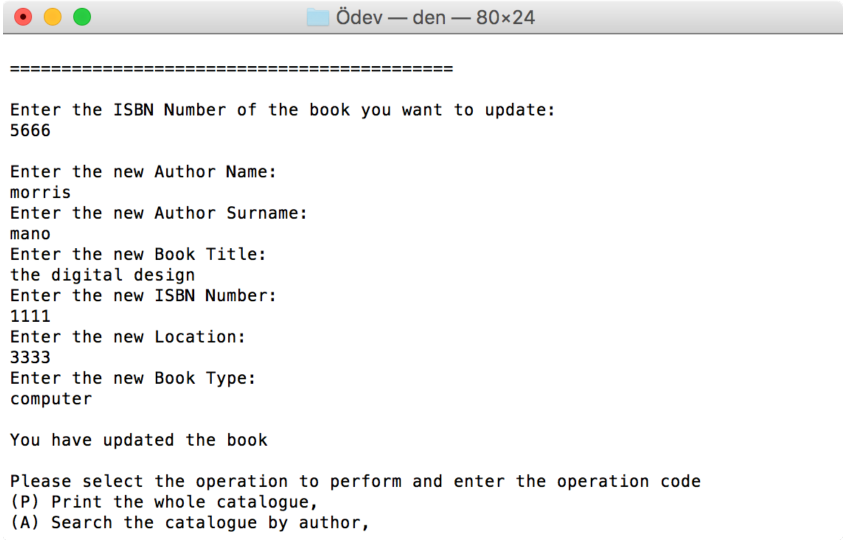
```
Can-MacBook-Pro:Ödev canaltinigne$ ./den

Please select the operation to perform and enter the operation code
(P) Print the whole catalogue,
(A) Search the catalogue by author,
(T) Search the catalogue by book type,
(I) Insert a new book record,
(U) Update an existing book record,
(R) Remove a book record,
(E) Exit the program.
Your Selection is: i

Enter the Author Name:
can
Enter the Author Surname:
altinigne
Enter the Book Title:
deneme
Enter the ISBN Number:
1222
Enter the Location:
2111
Enter the Book Type:
deneme
```



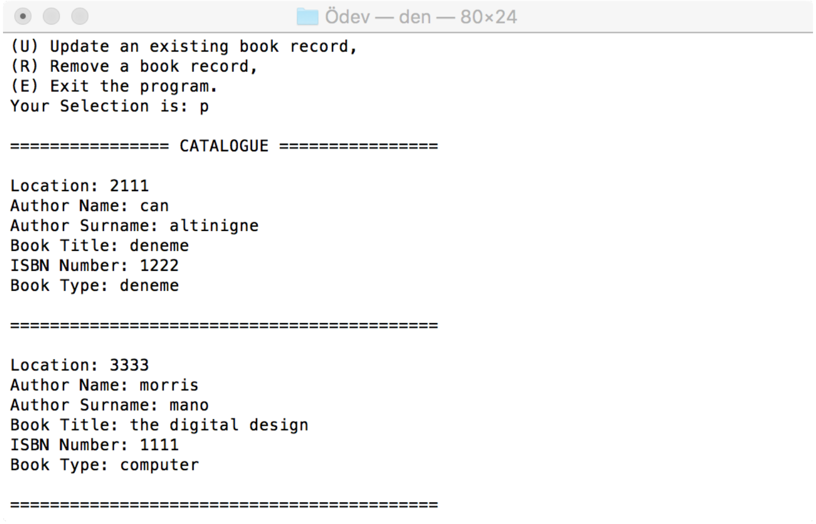
```
(R) Remove a book record,
(E) Exit the program.
Your Selection is: p

================ CATALOGUE ================

Location: 2111
Author Name: can
Author Surname: altinigne
Book Title: deneme
ISBN Number: 1222
Book Type: deneme

===========================================

Please select the operation to perform and enter the operation code
(P) Print the whole catalogue,
(A) Search the catalogue by author,
(T) Search the catalogue by book type,
(I) Insert a new book record,
(U) Update an existing book record,
(R) Remove a book record,
(E) Exit the program.
Your Selection is:
```



```
ALL BOOKS IN THE CATALOGUE

================ CATALOGUE ================

Location: 2111
Author Name: can
Author Surname: altinigne
Book Title: deneme
ISBN Number: 1222
Book Type: deneme

===========================================

Location: 2222
Author Name: andy
Author Surname: weir
Book Title: the martian
ISBN Number: 5666
Book Type: sci fi

===========================================

Enter the ISBN Number of the book you want to update:
```



```
===========================================

Enter the ISBN Number of the book you want to update:
5666

Enter the new Author Name:
morris
Enter the new Author Surname:
mano
Enter the new Book Title:
the digital design
Enter the new ISBN Number:
1111
Enter the new Location:
3333
Enter the new Book Type:
computer

You have updated the book

Please select the operation to perform and enter the operation code
(P) Print the whole catalogue,
(A) Search the catalogue by author,
```



```
(U) Update an existing book record,
(R) Remove a book record,
(E) Exit the program.
Your Selection is: p

================ CATALOGUE ================

Location: 2111
Author Name: can
Author Surname: altinigne
Book Title: deneme
ISBN Number: 1222
Book Type: deneme

===========================================

Location: 3333
Author Name: morris
Author Surname: mano
Book Title: the digital design
ISBN Number: 1111
Book Type: computer

===========================================
```